Práctica 1: Búsquedas con trayectorias simples
Selección de características
Primero el mejor, enfriamiento simulado y búsqueda tabú básica

Alejandro García Montoro
76628233F, agarciamontoro@correo.ugr.es

Grupo de los viernes a las 17.30

Curso 2015 - 2016

Índice

1. Descripción del problema

La selección de características es una técnica muy usada en problemas de aprendizaje automático.

El aprendizaje automático, visto de una forma muy general, tiene como objetivo clasificar un conjunto de objetos —modelador por una serie de atributos— en clases.

Esta clasificación se aprende desde los datos, pero la selección de los atributos que definen la modelización del objeto puede no ser la más apropiada: en ocasiones hay atributos superfluos o demasiado ruidosos que sería conveniente eliminar. Además, cuantos menos atributos definan un objeto, más rápido y preciso será el aprendizaje. Es aquí entonces donde aparece la pregunta que guia todo este trabajo: ¿cómo identificar los atributos que mejor aprendizaje promueven?

La respuesta a esta pregunta pasa por la selección de características, cuyo objetivo es reducir la definición de un objeto a una serie de características que faciliten el aprendizaje.

La idea es entonces la siguiente: dado un conjunto de m objetos definidos por un conjunto C de n características y considerada un modelo de aprendizaje f que intenta aprender la clasificación de estos objetos encontrar el subconjunto $C' \subset C$ que maximiza del modelo f.

Así, vemos claramente que el tamaño de caso de nuestro problema es n, el número de características, y que el objetivo está bien definido: eliminar aquellas características que o bien empeoren la bondad de f o bien sean innecesarias.

Con todos estos elementos definidos, podemos pasar a analizar las metaheurísticas consideradas.

2. Metaheurísticas

2.1. Introducción

Los algoritmos considerados para resolver el problema son los siguientes:

- *Best first local search*
- *Simulated annealing*
- *Short-term memory tabu search*

Además, compararemos estas metaheurísticas con el algoritmo voraz *Sequential forward selection*.

Estas tres metaheurísticas reúnen las condiciones necesarias para resolver el problema: el espacio de soluciones de nuestro problema puede ser analizado mediante las estructuras de generación de vecinos y los criterios de aceptación que utilizan estos algoritmos. Veamos con un poco más de detalle los aspectos comunes a las metaheurísticas implementadas:

Datos de entrada

Todos los algoritmos considerados reciben un conjunto de entrenamiento cuyos objetos tienen la siguiente estructura:

$$(s_1, s_2, \ldots, s_n, c)$$

donde $(s_1, s_2, \ldots, s_n)$ es el conjunto de valores de los atributos que definen el objeto y c la clase a la que pertenece.

Esquema de representación

El espacio de soluciones S de nuestro problema es el conjunto de todos los vectores s de longitud n —el número de características— binarios; es decir:

$$S = \{s = (s_1, s_2, \ldots, s_n) / s_i \in \{0, 1\}\ \forall i = 1, 2, \ldots, n\}$$

La posición i-ésima de un vector $s \in S$ indicará la inclusión o no de la característica i-ésima en el conjunto final C'.

Función objetivo

La finalidad de las metaheurísticas será maximizar la función objetivo siguiente:

$$\begin{aligned} f &: S \to [0, 100] \\ s &\mapsto f(s) = \text{Acierto del 3-NN sobre s} \end{aligned}$$

$f(s)$ es, por tanto, la tasa de acierto del clasificador 3-NN producido a partir de la solución s.

El clasificador 3-NN es una particularización del clasificador k-NN, que mide la distancia de la instancia considerada a todos los demás objetos en el conjunto de datos de entrenamiento y le asigna la clasificación mayoritaria de entre los k vecinos más cercanos; esto es:

Pseudocódigo 1 Clasificador k-NN

```
1: function k-NN(instance, trainingData)
2:     distances ← euclideanDistance(instance, trainingData)
3:     neighbours ← getClosestNeighbours(distances)
4:     classification ← mostVotedClassification(neighbours)
5:     return classification
```

Entorno de soluciones

Dada una solución $s \in S$, el entorno de soluciones vecinas a s es el conjunto

$$E(s) = \{s' \in S / s' - s = (0, \ldots, 0, \underbrace{1}_{i}, 0, \ldots, 0), i \in \{1, 2, \ldots, n\}\}$$

es decir, $E(s)$ son las soluciones que difieren de s en una única posición. Es evidente entonces que el conjunto $E(S)$ tiene siempre exactamente cardinal igual a n.

El operador de generación de vecino de la solución s es entonces como sigue:

Pseudocódigo 2 Operador de generación de vecino

```
1: function FLIP(solution, feature)
2:     s' ← solution
3:     s'[feature] ← (s'[feature] + 1) mod 2
4:     return s'
```

Criterios de parada

Aunque los criterios de parada dependerán de la metaheurística considerada —en general se parará cuando no se encuentre mejora en el entorno—, en todos los algoritmos pararemos necesariamente tras llegar a las 15000 evaluaciones con el clasificador 3-NN sobre las soluciones generadas.

2.2. Búsqueda local primero el mejor

El primer algoritmo considerado es una búsqueda local de primero el mejor muy sencilla. El pseudocódigo de todo el procedimiento es el siguiente:

Pseudocódigo 3 Búsqueda local primero el mejor

```
1: function BESTFIRST(train, target)
2:     s ← genInitSolution()
3:     besScore ← score(s, train, target)
4:     improvementFound ← True
5:     while improvementFound do
6:         improvementFound ← False
7:         for f ← genRandomFeature(s) do          ▷ Without replacement
8:             s'← genNeighbour(s,f)
9:             score ← score(s', train, target)
10:            if score > bestScore then
11:                bestScore ← score
12:                s ← s'
13:                improvementFound ← True
14:                break
15:    return s, bestScore
```

El método de exploración del entorno es el siguiente: dada una solución s, escogemos una característica al azar, aplicamos el operador $flip$ para obtener una solución vecina y medimos su bondad; si es mejor que s, nos quedamos con ella como mejor solución y volvemos a empezar; si no, tomamos otra característica al azar —sin repetir— y seguimos el proceso.

Pararemos el algoritmo si y sólo si, al haber explorado el entorno completo de la solución actual, ninguna de las soluciones vecinas es mejor. Estaremos entonces ante un máximo —probablemente local— y el algoritmo no puede mejorar la solución.

2.2.1. Enfriamiento simulado

La metaheurística de enfriamiento simulado es un ejemplo de estrategia de búsqueda por trayectorias simples.

La idea de este algoritmo es mantener una variable de temperatura, de manera que cuando esta sea alta la diversificación en el entorno de búsqueda será muy amplia —podremos pasar a zonas peores, explorando así muchas zonas diferentes del espacio de búsqueda y evitando máximos locales— y conforme tiene a la temperatura final, se procede a una fase de intensificación sobre una parte del espacio.

En este caso, además, debemos almacenar siempre la mejor solución, de manera que aunque al final intensifiquemos sobre una zona pobre, si al principio la diversificación fue exitosa, tengamos más posibilidades de obtener una solución buena.

Antes de entrar en los detalles, veamos primero el pseudocódigo del procedimiento en general:

Pseudocódigo 4 Enfriamiento simulado

```
1: function SIMULATEDANNEALING(train, target)
2:     s ← genInitSolution()
3:     bestSolution ← s
4:     bestScore ← score(s, train, target)
5:     currentScore ← bestScore
6:     t ← t_0
7:     while t > t_f and neighboursAccepted > 0 and eval < 15000 do
8:         neighboursAccepted ← 0
9:         while not cooling needed do
10:            f ← genRandomFeature(s)                ▷ With replacement
11:            s'← genNeighbour(s,f)
12:            newScore ← score(s', train, target)
13:            Δ = currentScore − newScore
14:            if Δ < 0 or acceptWorseSolution = True then
15:                currentScore ← newScore
16:                acceptedNeighbourgs++
17:                if currentScore > bestScore then
18:                    bestScore ← currentScore
19:                    bestSolution ← s
20:        t ← coolingScheme(t)
21:    return s, bestScore
```

En este algoritmo hay tres cuestiones que debemos detallar: la generación de la temperatura inicial, la condición que se debe de cumplir para proceder al enfriamiento, y la determinación de la aceptación de una solución peor que la actual.

	WDBC				Movement Libras				Arrythmia			
Particiones	%Clas. in	%Clas. out	%Red.	T	%Clas. in	%Clas. out	%Red.	T	%Clas. in	%Clas. out	%Red.	T
Partición 1-1	96,4789	98,2456	0	0,3155	66,1111	73,8889	0	0,2082	65,625	63,4021	0	0,3232
Partición 1-2	95,7895	95,0704	0	0,3099	69,4444	67,7778	0	0,2072	64,433	65,1042	0	0,3188
Partición 2-1	95,0704	96,4912	0	0,3089	68,8889	73,3333	0	0,2083	63,5417	67,5258	0	0,3266
Partición 2-2	97,193	96,831	0	0,3102	63,3333	73,8889	0	0,2079	63,9175	62,5	0	0,318
Partición 3-1	95,4225	97,193	0	0,3091	66,1111	67,2222	0	0,2073	63,5417	63,9175	0	0,3221
Partición 3-2	96,8421	97,1831	0	0,3158	70,5556	72,7778	0	0,2072	62,3711	64,0625	0	0,3187
Partición 4-1	95,7746	97,193	0	0,3093	69,4444	70,5556	0	0,2125	61,9792	64,9485	0	0,3271
Partición 4-2	95,7895	96,831	0	0,3104	67,7778	74,4444	0	0,2073	61,8557	65,1042	0	0,3192
Partición 5-1	96,831	94,0351	0	0,3089	66,6667	69,4444	0	0,2076	63,0208	64,433	0	0,3232
Partición 5-2	95,0877	96,831	0	0,3119	71,6667	73,8889	0	0,2082	61,8557	64,5833	0	0,3192
Medias	96,0279	96,5904	0	0,311	68	71,7222	0	0,2082	63,2141	64,5581	0	0,3216

	WDBC				Movement Libras				Arrythmia			
Particiones	%Clas. in	%Clas. out	%Red.	T	%Clas. in	%Clas. out	%Red.	T	%Clas. in	%Clas. out	%Red.	T
Partición 1-1	98,2394	93,6842	86,6667	37,9223	80	70	85,5556	199,9326	83,3333	76,2887	97,482	499,5912
Partición 1-2	96,4912	97,5352	83,3333	44,4401	77,7778	70	88,8889	159,6696	79,3814	68,2292	97,1223	577,516
Partición 2-1	98,2394	93,3333	80	50,9161	72,2222	61,6667	92,2222	117,551	78,125	71,134	98,5612	302,55
Partición 2-2	97,193	95,0704	86,6667	37,6991	74,4444	66,1111	88,8889	159,1287	77,3196	67,1875	97,482	538,2686
Partición 3-1	97,5352	94,386	83,3333	44,3802	69,4444	64,4444	92,2222	117,3026	84,8958	72,1649	95,6835	762,5215
Partición 3-2	98,2456	95,7746	86,6667	37,6586	72,7778	63,8889	85,5556	199,3051	78,866	69,2708	97,8417	390,8462
Partición 4-1	96,1268	95,4386	90	30,5451	71,6667	75	90	145,5246	76,0417	69,5876	98,2014	330,9537
Partición 4-2	97,5439	93,662	83,3333	44,5188	73,8889	66,6667	90	147,9849	75,2577	69,2708	98,9209	221,1512
Partición 5-1	98,5915	95,0877	80	50,8462	74,4444	65,5556	88,8889	159,8845	82,2917	67,5258	97,8417	355,9443
Partición 5-2	97,8947	95,0704	83,3333	44,5186	72,7778	65	90	145,2093	74,7423	66,1458	98,9209	204,8205
Medias	97,6101	94,9042	84,3333	42,3445	73,9444	66,8333	89,2222	155,1493	79,0255	69,6805	97,8058	418,4163

	WDBC				Movement Libras				Arrythmia			
Particiones	%Clas. in	%Clas. out	%Red.	T	%Clas. in	%Clas. out	%Red.	T	%Clas. in	%Clas. out	%Red.	T
Partición 1-1	98,9437	94,7368	43,3333	19,2276	76,6667	70,5556	58,8889	24,0088	66,6667	64,9485	49,6403	113,5048
Partición 1-2	96,8421	97,5352	43,3333	25,5444	68,3333	74,4444	45,5556	30,4084	67,0103	63,0208	50,7194	145,5799
Partición 2-1	97,8873	95,4386	50	16,0968	71,1111	73,8889	50	31,2889	68,2292	64,433	47,482	219,7524
Partición 2-2	97,5439	95,4225	50	27,4419	73,3333	72,7778	43,3333	65,4793	67,5258	62,5	50,3597	251,5668
Partición 3-1	96,831	94,7368	43,3333	11,6721	75,5556	69,4444	52,2222	62,4497	67,1875	63,4021	48,5612	187,7998
Partición 3-2	97,8947	96,1268	26,6667	13,6743	70	68,3333	53,3333	44,759	68,0412	65,1042	47,8417	250,5469
Partición 4-1	97,1831	95,4386	63,3333	18,4396	72,2222	73,8889	51,1111	42,4045	74,4792	63,9175	54,3165	246,3999
Partición 4-2	95,7895	94,7183	60	10,0989	69,4444	73,3333	40	30,2926	66,4948	64,5833	47,1223	197,383
Partición 5-1	97,1831	95,7895	43,3333	13,352	72,7778	67,2222	52,2222	36,6807	68,2292	61,3402	53,9568	199,7321
Partición 5-2	96,8421	96,831	56,6667	9,0114	71,6667	67,7778	47,7778	23,7236	68,5567	61,9792	47,8417	254,1432
Medias	97,294	95,6774	48	16,4559	72,1111	71,1667	49,4444	39,1495	68,2421	63,5229	49,7842	206,6409

	WDBC				Movement Libras				Arrythmia			
Particiones	%Clas. in	%Clas. out	%Red.	T	%Clas. in	%Clas. out	%Red.	T	%Clas. in	%Clas. out	%Red.	T
Partición 1-1	98,9437	94,0351	43,3333	85,231	75,5556	75	43,3333	167,9859	77,6042	64,9485	53,9568	839,9968
Partición 1-2	96,8421	97,1831	56,6667	84,2392	76,1111	72,7778	52,2222	164,5101	70,6186	67,1875	50,7194	833,381
Partición 2-1	97,8873	92,2807	40	84,4377	72,2222	66,6667	53,3333	165,8923	78,125	68,5567	54,3165	802,2534
Partición 2-2	97,5439	96,831	56,6667	84,3516	71,6667	70	48,8889	168,0965	71,134	61,4583	51,0791	736,6535
Partición 3-1	97,5352	95,0877	50	84,157	77,2222	73,3333	53,3333	168,5341	77,0833	65,4639	48,2014	696,282
Partición 3-2	98,9474	96,1268	43,3333	84,9611	74,4444	63,8889	48,8889	172,075	73,1959	65,1042	55,036	660,9853
Partición 4-1	97,8873	94,7368	63,3333	83,6998	81,1111	67,7778	48,8889	167,2869	74,4792	63,9175	49,6403	667,9511
Partición 4-2	97,8947	95,0704	33,3333	85,8251	77,7778	72,2222	52,2222	163,6381	75,2577	68,2292	53,5971	667,1272
Partición 5-1	98,2394	94,7368	43,3333	84,4193	78,3333	70	52,2222	169,1234	72,9167	63,9175	61,1511	655,1922
Partición 5-2	97,8947	94,7183	40	85,0379	69,4444	73,3333	51,1111	166,7657	72,6804	65,625	42,8058	690,5834
Medias	97,9616	95,0807	47	84,636	75,3889	70,5	50,4444	167,3908	74,3095	65,4408	52,0504	725,0406

	WDBC				Movement Libras				Arrythmia			
Particiones	%Clas. in	%Clas. out	%Red.	T	%Clas. in	%Clas. out	%Red.	T	%Clas. in	%Clas. out	%Red.	T
Partición 1-1	99,2958	95,7895	40	4,300,7671	73,3333	68,3333	60	3,286,9409	73,4375	62,3711	51,7986	4,227,4062
Partición 1-2	97,8947	97,1831	60	4,232,4826	80,5556	70,5556	53,3333	2,776,5594	75,2577	68,75	50,7194	3,770,462
Partición 2-1	98,9437	94,386	53,3333	4,236,1506	75	70,5556	56,6667	2,838,2285	72,3958	64,433	57,1942	3,573,1368
Partición 2-2	98,9474	96,4789	50	4,264,8892	79,4444	71,1111	63,3333	2,734,6227	75,2577	63,5417	51,0791	3,695,1107
Partición 3-1	98,9437	94,0351	60	4,216,179	79,4444	75	57,7778	2,729,8555	77,0833	65,9794	55,7554	3,582,3163
Partición 3-2	99,2982	94,7183	46,6667	4,282,2213	79,4444	72,2222	63,3333	2,759,1975	77,8351	70,3125	55,7554	3,628,5453
Partición 4-1	98,5915	96,1404	50	4,268,7189	82,2222	76,1111	52,2222	2,781,284	73,9583	64,433	53,5971	3,631,7597
Partición 4-2	98,9474	93,3099	50	4,287,3348	75	67,2222	63,3333	2,762,6962	76,8041	68,2292	58,2734	3,615,0633
Partición 5-1	99,2958	94,386	43,3333	4,237,2314	75,5556	65,5556	65,5556	2,729,8462	72,9167	61,8557	48,2014	3,666,4759
Partición 5-2	98,2456	96,1268	46,6667	4,252,624	78,8889	71,6667	55,5556	2,763,403	74,2268	64,0625	54,3165	3,637,0293
Medias	98,8404	95,2554	50	4,257,8599	77,8889	70,8333	59,1111	2,816,2634	74,9173	65,3968	53,669	3,702,7306